Argo UML Tool
Tutorial

Peter King

1 Introduction

Argo is a UML design tool being developed by University of Southern California and others. It supports development of UML designs and can generate outline code from the diagrams. The manual (a little bit patchy) is on line on the DIS. From the Departmental Home page go to Departmental Information System → Computing Facilities → User Guides → Argo Guide.

2 Example

This tutorial takes you through the development of use case diagrams and class diagrams for the video rental store example.

> Develop a software system to support a video rental store. The system should automate the process of renting tapes and having tapes returned by borrowers. Calculation of all charges concerning rental will be automated by the system, including extra charges for overdue items. The rental may be charged in advance, when the tape is rented, or it may be charged when the tape is returned. Borrowers are known to the system by a unique identifier read from a bar code on their membership card. The borrower's card identifies them as having a particular class of membership, which may restrict the number of videos they mare allowed to borrow simultaneously, or prevent them borrowing videos that have a particular censorship rating. The video tapes are also uniquely identified by a bar code. Bar codes are read with a laser scanner. There may be multiple copies of any video title, but each will have a different bar code. Tapes also have a running time, a price code, and a rating imposed by the local authorities that affects the age of person allowed to rent that tape. The system must allow for tapes to be added and removed from the shop's stock. Customers must be able to change their name and address details. Staff must be able to find out if a title is available or if all copies are on loan.

3 Use Case Diagrams

Start Argo from any terminal window by typing

```
argo
```

After a short delay a small window appears shown in Figure 1. The progress bar at the foot of the window will show the names of internal packages being loaded, and when complete, the window disappears and is replaced by the main window, shown in Figure 2 (the dimensions may be slightly different, but all components should be visible). When iconified, argo displays a very small folded paper boat as an icon[1].

Making Project Browser

Figure 1: Splash Screen for ArgoUML

- You will observe that the bottom left of the window has a tree structured list, with three "directory" type icons. The heading says By Priority 2 Items. This can be seen if Figure 3. Clicking on the key icon next to Medium shows the two items Add Elements to Package untitledmodel and Revise Package Name untitledmodel.

  Clicking on Revise Package Name untitledModel causes the Details pane to display an explanation of why the name is poor, and allows you to change it. Click next at the bottom of that area, and a "Wizard" appears, with the text that needs changing, as shown in Figure 4. Replace untitledmodel (Argo will have already removed the capital letter!) with videostore and then click on the Finish button.

argo is continually monitoring the changes you make to your models and advising you of any problems that they may cause. This criticism of your model is controlled by the Critique pulldown menu.

- To create a use case diagram, click on the Usecase Diagram 1 in the navigation pane. The icons on the buttons along the top of the editing pane

[1] In Greek mythology, *Argo* was the name of Jason's boat when he was seeking the Golden Fleece. Perhaps this icon is some sort of joke or pun!

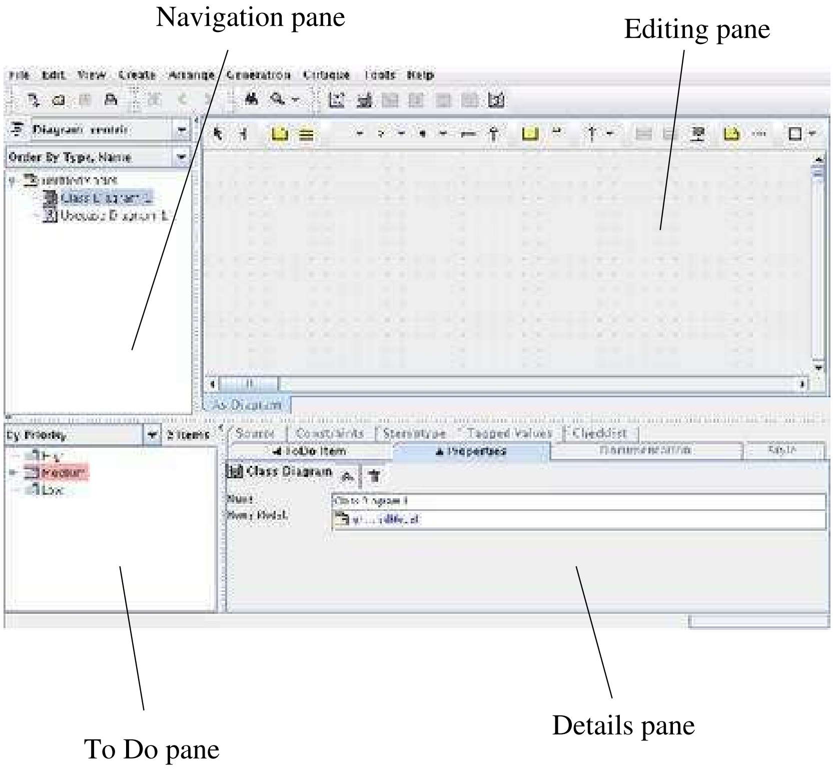


Figure 2: argo: Screen Layout

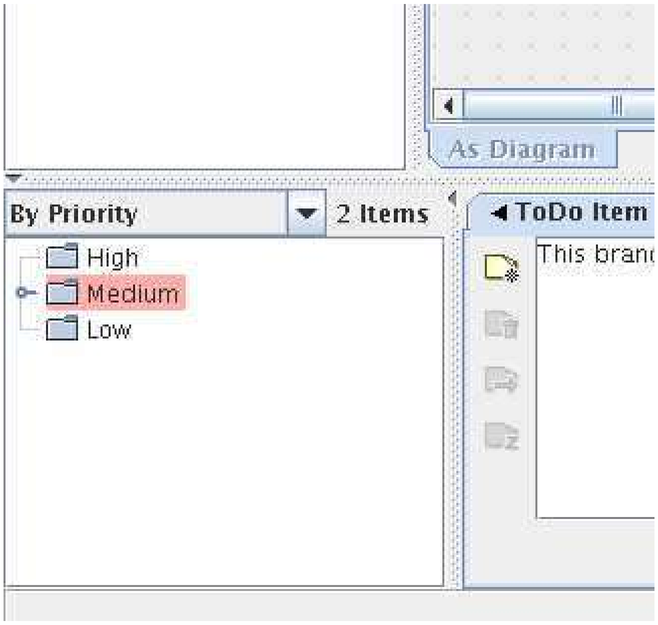
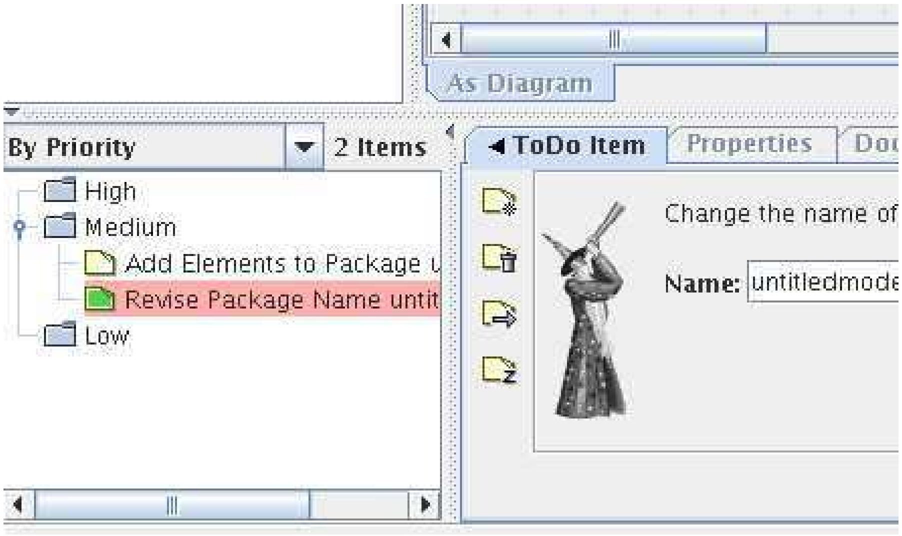


Figure 3: To Do pane



Change the name of

Name: untitledmode

Figure 4: Wizard correcting To Do Item

will change to reflect use cases, specifically including actors and use-case icons.

Click on the actor button, and place the actor in the editing pane. It will start with no name indicated by a small wavy red line. Typing will add a name - say Customer. The red line disappears when you click away from the actor. See Figure 5

- Add a use case, by clicking on the button with the use case icon and placing the ellipse in the editing pane. The wavy red line is to remind you that the use case needs a name. When you give it a name (borrow), you will find two new ToDo items have appeared, asking you to add Associations to the Customer and to the borrow use case.

- When you add an actor or a use case, a small association icon appears as a handle to the left and right of the actor or use case symbol. To add an association, hold down the mouse button in the handle, and drag it to the actor or use case for which an association is required.

- You can continue to add use cases, actors and associations. As they are added, you can either deal with the naming as each component is added, or you can wait and deal with all the names after the diagram is complete.

  Add a use case for borrow, return and query with associations with the customer and the staff member.

  As components are added, whether use cases, actors, or associations, they also appear in the navigation pane.

  If you want to delete a component, it is easiest to click on that component in the *navigation* pane, and the use the Edit menu from the menu bar.

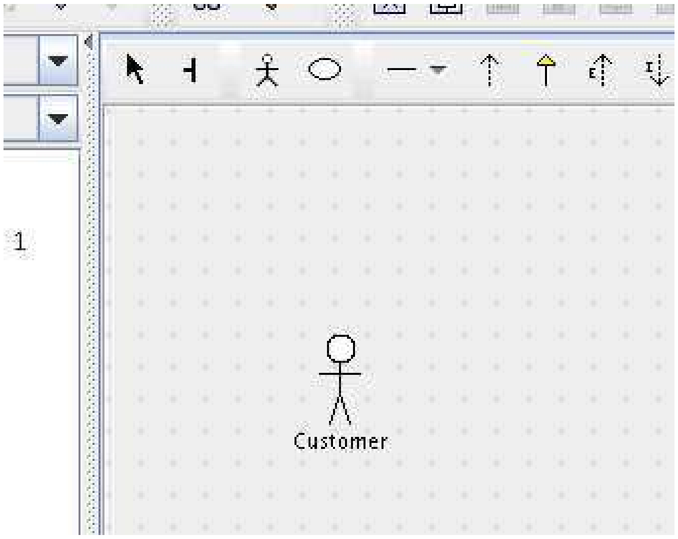


Figure 5: Actor

Beware, using the DELETE key will delete the component from the diagram only, not from the model. To delete a component completely from the model, use the menu.

- After adding multiple elements to a diagram, you may want to tidy it up somewhat, for example to move the use case ellipses so that they are all in the same vertical line, use the broom tool on the drawing toolbar ⊣ is its icon. It sweeps all before it, pushing any components, but keeping their associations intact. Assume you want to align all your use cases with the rightmost one. Click on the icon in the tool bar to select the broom, and move to a point to the left of the use cases, but to the right of the actor. Drag the mouse button to the right, and a vertical bar will appear, sweeping all components with it. If you need to make the bar longer, after you have started dragging the mouse, any vertical movement of the mouse will extend the bar in that direction.

  The direction of the sweeping is set by the first direction in which the mouse is dragged.

- It is probably worth giving the use case diagram a name, by clicking on it in the navigation pane, and then choosing the properties tab in the details pane. Name this one "Customer Use Cases".

- Comments (notes in UML terminology) can be added to actors, use cases, or to the associations between them by clicking on the component, and then clicking on the note button on the diagram editing row of buttons.

- The diagram can be saved as a graphic by choosing the File menu from the menu bar, and then Export Graphics... Graphics can be saved in Postscript, PNG, SVG, or GIF format and its file name will default to the name givne to the diagram.

It is worth saving your work from time to time. This is done in the File menu. I found that occasionally, argo would delete things I wasn't expecting, (probably my inexperience), but there doesn't seem to be an 'undo' for deletions! The diagrams and associated UML model is saved in a file of type `.zargo`. Previously saved projects can be reloaded from the File menu, or using the "open-folder" icon on the main toolbar.

4 Class Diagrams

Class diagrams are created by selecting Create menu from the menu, using the New Class Diagram icon from the toolbar, or (in the case of a new project), selecting the untitled Class Diagram 1 in the navigation pane.

- Creation of the classes happens using the class button. This is the button at the left hand end of the editing pane tool bar with the rectangle split into three horizontal rectangles, like a class diagram. In general, to find out what a button does, leave the mouse over the button for a few moments and a description will pop-up.

- As classes are added more to do items will appear. Associations are added using the associations button, or just by grabbing one of the association handles on a class and dragging the mouse to another class. Adding aggregation associations (with the diamond end) involves clicking the appropriate button in the toolbar, then the handle on the class which is to have the diamond, and dragging to the target class.

- Associations can have their multiplicities set by selecting the association, and then clicking the Properties tab in the details pane. Either end of the association can be clicked on at the right hand end of the pane. Once the appropriate end has been clicked, there are text boxes to select the multiplicity, and also the name which will be used for any code generated.

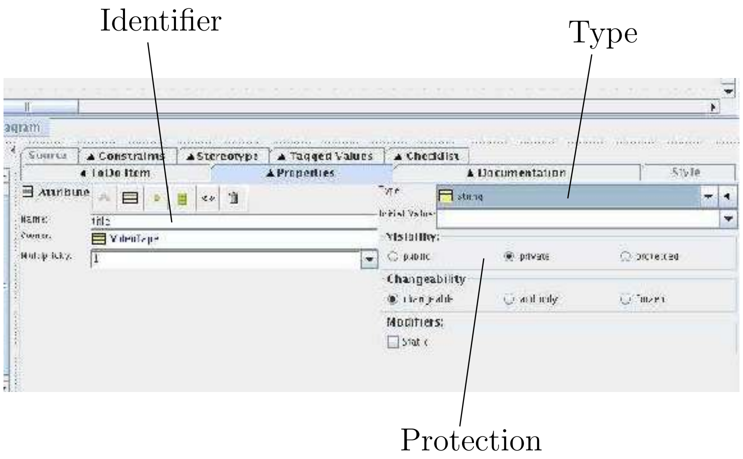
- Any component can be selected in the navigation pane and its properties edited.

  All the classes, associations, etc. that you add to the diagram are also added to the navigation pane.

- Adding attributes and operations to classes is done with the buttons at the top of the editing area, that have the corresponding area of the class diagram blacked out. These buttons are only available when a class has been selected as the current object.

- To add a generalisation of a class, use the Generalization tool from the button bar, and drag *from* the class you wish to be a specialisation *to* the class that is the generalisation.

  The generalisation will be added as a straight line from one class to the other — to make the generalisation a set of straight lines, drag a point in the line to the desired position.



Name: title
Owner: VideoTape
Multiplicity: 1
Type: string
Initial Value:
Visibility: ○ public ● private ○ protected
Changeability: ● changeable ○ addOnly ○ frozen
Modifiers: □ Static

Figure 6: Attribute Properties

4.1 Class Attributes

Attributes are added to a class either by selecting the appropriate To Do item, or by selecting the button from the tool bar. The Details Pane will be replaced with Properties tab, as shown in Figure 6.

The identifier is entered in the appropriate field, and then the type of the attribute can be set. The pulldown menu for types includes standard Java types such as `String`, but unfortunately, it also adds them to the To Do pane! The access to the attribute can also be specified.[2]

4.2 Class Operations

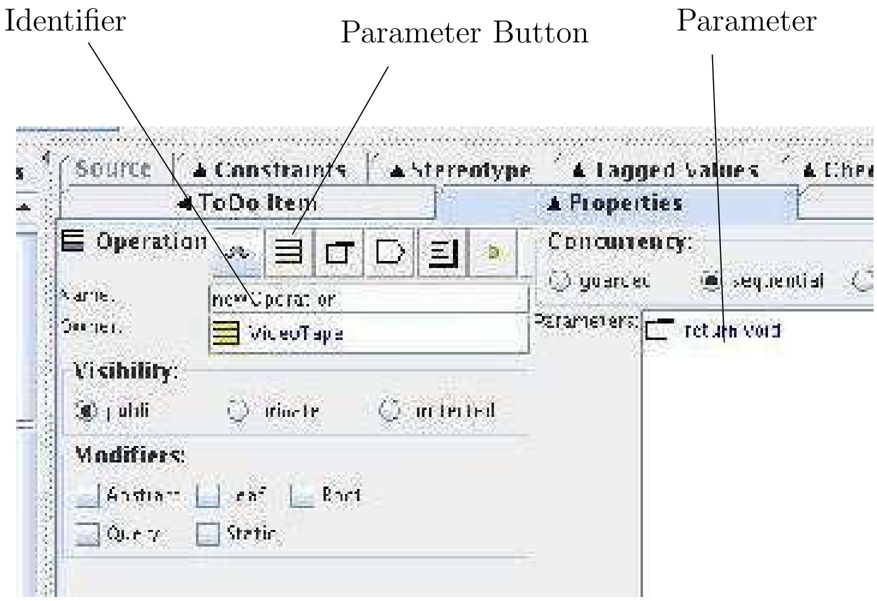
Operations are what UML calls Java methods. When New Operation is selected, either from the menu or the icon toolbar, the Details Pane will be replaced with Properties tab, as shown in Figure 7. Once again the identifier of the operation is specified in the appropriate field. Modifying the type returned by the operation is more awkward. Initially, the only parameter listed in the area is the return parameter. Double clicking on it's entry in the *Parameters* list brings up a similar screen to Figure 6 that can be used to set the type of the return value.

Other parameters can be added to the method by clicking on the parameter button, giving them identifiers and types in the same way as attributes and the return value. Argo will allow you to specify multiple return parameters - not all languages are like Java.

4.3 Multiple Diagrams

For many systems, a single class diagram will be too complicated to show every association. It is better to create a nubmer of different class diagrams, each concentrating on one aspect of the system. When a class that has been already defined is needed in a second or subsequent diagram, drag its icon from the navigation pane into the new diagram.

[2] However, see Section 5 for how to remove these items.



Name: newOperation
Owner: VideoTape
Visibility: ● public ○ private ○ protected
Modifiers: □ Abstract □ Leaf □ Root □ Query □ Static
Concurrency: ○ guarded ● sequential
Parameters: return void

Figure 7: Operation Properties

4.4 Presentation

This tab in the details pane allows the designer to specify the colours and line thicknesses for the graphic - as an aid to the reader!

5 To Do Items

To Do items can be added to any component you choose by clicking the To Do tab in the details pane. A new item is added by clicking on the topmost of the 4 icons at the left of the details pane.

Items can also be deleted, by highlighting the item in the To Do pane, and then choosing the dustbin icon from the left of the details pane.

6 Collaboration Diagrams

Collaboration diagrams require a class to be selected initially, before they can be drawn. Some experimentation should allow you to add messages to associations in the collaboration. If typing in a field in the details pane fails to have any effect, try right clicking the mouse in the field. It may produce a pulldown list of alternatives.

7 Activity Diagrams

Activity diagrams can only be drawn once a use case or a class has been highlighted.

8 State Diagrams

State diagrams can only be drawn once a class has been highlighted in a class diagram. Drawing simple states, initial and terminal states and transitions between them is straightforward. Adding guards, triggers and actions is done in the details pane.

9 Sequence Diagrams

Sequence diagrams can only be drawn once a class has been highlighted in a class diagram. Objects and actors are both represented by *classifiers*, the box icon. Adding one to the diagram also includes the dotted timeline down the diagram. Messages between the timelines are added by selecting the type of message - call, return, create, or destroy, and then dragging from the source to the target.